

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently amended) A method for starting a group of enterprise servers comprising:

comparing binaries and/or configuration settings stored within a local file system of each enterprise server with binaries and/or configuration settings stored within [[the]] a central database accessible to the group of enterprise servers to identify any binaries and/or configuration settings stored within the local file system which are out-of-date as compared to the binaries and/or configuration settings stored within the central database;

if the binaries and/or configuration settings stored within the local file system are out-of-date as compared to the binaries and/or configuration settings stored within the central database, then updating the binaries and/or configuration settings from the central database to the local file system prior to starting each enterprise server; and

starting each enterprise server using the updated binaries and/or configuration settings.

2. (Original) The method as in claim 1 further comprising:

generating a list of servers within the group to be started based on server layout information retrieved from the central database, the server layout information uniquely identifying each server in the group and/or parameters associated with each server in the group.

3. (Original) The method as in claim 2 wherein said layout information is defined by a configuration hierarchy stored within a hierarchical data object in the central database.

4. (Original) The method as in claim 3 wherein the hierarchical data object comprises a global sub-hierarchy and a non-global sub-hierarchy, the global sub-hierarchy containing configuration data and binaries associated with all of the sever nodes and the non-global sub-hierarchy containing the layout information, configuration data and binaries associated with particular sever nodes.

5. (Original) The method as in claim 1 wherein the group of enterprise servers comprises an instance of enterprise servers.

6. (Original) The method as in claim 5 wherein the instance of enterprise servers comprises at least one dispatcher and two or more server nodes.

7. (Original) The method as in claim 1 wherein the servers within the group comprise Java 2 Enterprise Edition ("J2EE") servers.

8. (Original) A system for storing a group of enterprise servers comprising:

a central database for storing binaries and configuration settings associated with the group of enterprise servers;

bootstrap logic to compare binaries and/or configuration settings stored within a local file system of each enterprise server with the binaries and/or configuration

settings stored within the central database to identify any binaries and/or configuration settings stored within the local file system which are out-of-date,

wherein if the binaries and/or configuration settings stored within the local file system are out-of-date, then the bootstrap logic updates the binaries and/or configuration settings from the central database to the local file system prior to starting each enterprise server; and

startup and control logic to start each enterprise server using the updated binaries and/or configuration settings.

9. (Original) The system as in claim 8 wherein said bootstrap logic comprises node-specific bootstrap logic, the system further comprising:

group bootstrap logic to generate a list of servers within the group to be started based on server layout information retrieved from the central database, the server layout information uniquely identifying to the startup and control logic each server in the group and/or parameters associated with each server in the group.

10. (Original) The system as in claim 9 wherein said layout information is defined by a configuration hierarchy stored within a hierarchical data object in the central database.

11. (Original) The system as in claim 10 wherein the hierarchical data object comprises a global sub-hierarchy and a non-global sub-hierarchy, the global sub-hierarchy containing configuration data and binaries associated with all of the servers and the non-global sub-hierarchy containing the layout information and configuration data and binaries associated with specified individual servers.

12. (Original) The system as in claim 8 wherein the group of enterprise servers comprises an instance of enterprise servers.

13. (Original) The system as in claim 12 wherein the instance of enterprise servers comprises at least one dispatcher and two or more sever nodes.

14. (Original) The system as in claim 8 wherein the servers within the group comprise Java 2 Enterprise Edition ("J2EE") servers.

15. (Currently amended) A system for preparing a group of servers for startup comprising:

first bootstrap logic to retrieve layout data from a central database, the layout data identifying the servers within the group and configuration parameters to be used for the servers within the group, the first bootstrap logic generating a data object identifying the servers and/or configuration parameters; and

second bootstrap logic to compare binaries and/or configuration settings stored within a local file system of each server with the binaries and/or configuration settings stored within [[the]] a central database accessible to the group of servers to identify any binaries and/or configuration settings stored within the local file system which are out-of-date as compared to the binaries and/or configuration settings stored within the central database,

wherein if the binaries and/or configuration settings stored within the local file system are out-of-date, then the bootstrap logic updates the binaries and/or configuration settings from the central database to the local file system prior to starting each server.

16. (Original) The system as in claim 15 further comprising: startup and

control logic to start each server identified in the data object using the updated binaries and/or configuration settings.

17. (Original) The system as in claim 15 wherein said layout data is defined by a configuration hierarchy stored within a hierarchical data object in the central database.

18. (Original) The system as in claim 17 wherein the hierarchical data object comprises a global sub-hierarchy and a non-global sub-hierarchy, the global sub-hierarchy containing configuration data and binaries associated with all of the servers and the non-global sub-hierarchy containing the layout information and configuration data and binaries associated with specified individual servers.

19. (Original) The system as in claim 15 wherein the group of servers comprises an instance of enterprise servers.

20. (Original) The system as in claim 19 wherein the instance of enterprise servers comprises at least one dispatcher and two or more sever nodes.

21. (Original) The system as in claim 15 wherein the servers within the group comprise Java 2 Enterprise Edition ("J2EE") servers.